



LABORATÓRIO DE PROGRAMAÇÃO – CK0215 2021.1 T01A

TRABALHO 4: DICIONÁRIO VIA ÁRVORE AVL (2021-08-18)

PROFESSOR: PABLO MAYCKON SILVA FARIAS

## 1 Resumo

Você deverá implementar em C++ um dicionário baseado em árvore AVL, conforme a interface apresentada mais à frente. A sua implementação será acoplada a um programa do professor, que realizará uma bateria de testes automáticos sobre as operações de dicionário e sobre a estrutura da árvore; caso seja detectado algum problema, o professor o informará para que seja feita uma correção (desde que dentro do prazo de submissão). A implementação será entregue ao professor através de tarefa no SIGAA.

**Atenção:** Colabore para o bom andamento da disciplina e do aprendizado de todos, não copiando códigos de outras pessoas nem compartilhando seu próprio código, nem mesmo em versão preliminar. Em caso de dificuldade sua ou com algum colega, fale com o professor.

## 2 Interface do Dicionário

Você deve implementar o dicionário a partir [desta interface](#), escrevendo os métodos solicitados e adicionando o que for necessário para isso.

O uso de um dicionário com a interface em questão deve ser imediato dadas as explicações feitas nas aulas da disciplina, mas de qualquer forma segue um exemplo de uso:

```
#include <iostream>
using std::cout;

int main ()
{
    DicioAVL<int,char> D;  int i;

    for (i = 48; i < 58; ++i) if (D.inserir(i, (char) i) == D.fim()) return 1;

    for (auto it = D.inicio(); it != D.fim(); ++it)
    {
        cout << "O código de '" << it.valor() << "' é " << it.chave() << '\n';
    }

    for (i = 48; i < 58; ++i)
    {
        auto it = D.buscar(i);  if (it == D.fim()) return 1; else D.remover(it);
    }
}
```

### 3 Compilação e Submissão da Solução

A sua implementação deverá ser escrita em C++17 padrão. Você deverá completar o arquivo da interface, renomeando-o para `solucao.cpp` e incluindo sua identificação no início do arquivo:

```
// Nome:      ...  
// Matrícula: ...
```

Eventuais comentários sobre a sua solução do trabalho podem também ser feitos no início do arquivo.

O professor incluirá o seu código num programa de testes, a ser compilado através de uma linha como:

```
g++ -Wall -Wextra -std=c++17 -pedantic -o programa main.cpp
```

A compilação do seu código deve ocorrer sem erros ou avisos.

Atente ao fato de que a sua solução para o trabalho consiste apenas na implementação do dicionário `DicioAVL`, e não em um programa inteiro. Naturalmente, você precisará testar o seu dicionário antes de submetê-lo, e portanto deverá escrever uma função `main` para fazer os testes; entretanto, para evitar conflito com o código do professor, você não deve enviar a sua função `main` no arquivo da solução do trabalho, ou então deve comentar tudo o que não seja a implementação do dicionário.

Você deve submeter sua solução para o trabalho através de tarefa a ser cadastrada pelo professor no SIGAA. O professor tentará testar a sua solução e dar um retorno no máximo no dia útil seguinte, de forma que você possa tentar corrigir sua implementação e ressubmetê-la, desde que dentro do prazo.

**Importante:** Não deixe para submeter a sua solução apenas no fim do prazo, pois, se isso acontecer, é bem possível que não haja tempo para o professor lhe dar um retorno e você corrigir a sua implementação.

Em caso de dúvida, contate o professor rapidamente.

**Observação:** Em princípio, você não deve usar a biblioteca padrão de C++ neste trabalho, exceto pelo `std::nothrow` de `<new>`. Caso você deseje utilizar algo a mais, contate o professor.

– Eu espero que você se divirta e aprenda. Bom trabalho! –